A Multilayer Interval Type-2 Fuzzy Extreme Learning Machine for the Recognition of Walking Activities and Gait Events using Wearable Sensors

Adrian Rubio-Solis[a], George Panoutsos[a], Carlos Beltran-Perez[a], Uriel Martinez-Hernandez[b]

[a]*Department of Automatic Control and Systems Engineering, University of Sheffield, UK, (email: a.rubiosolis@sheffield.ac.uk, g.panoutsos@sheffield.ac.uk, cbeltranperez1@sheffield.ac.uk)*
[b]*Department of Electronic and Electrical Engineering, University of Bath, UK, (email:u.martinez@bath.ac.uk)*

Abstract

In this paper, a novel Multilayer Interval Type-2 Fuzzy Extreme Learning Machine (ML-IT2-FELM) for the recognition of walking activities and Gait events is presented. The ML-IT2-FELM uses a hierarchical learning scheme that consists of multiple layers of IT2 Fuzzy Autoencoders (FAEs), followed by a final classification layer based on an IT2-FELM architecture. The core building block in the ML-IT2-FELM is the IT2-FELM, which is a generalised model of the Interval Type-2 Radial Basis Function Neural Network (IT2-RBFNN) and that is functionally equivalent to a class of simplified IT2 Fuzzy Logic Systems (FLSs). Each FAE in the ML-IT2-FELM employs an output layer with a direct-defuzzification process based on the Nie-Tan algorithm, while the IT2-FELM classifier includes a Karnik-Mendel type-reduction method (KM). Real data was collected using three inertial measurements units attached to the thigh, shank and foot of twelve healthy participants. The validation of the ML-IT2-FELM method is performed with two different experiments. The first experiment involves the recognition of three different walking activities: Level-Ground Walking (LGW), Ramp Ascent (RA) and Ramp Descent (RD). The second experiment consists of the recognition of stance and swing phases during the gait cycle. In addition, to compare the efficiency of the ML-IT2-FELM with other ML fuzzy methodologies, a kernel-based ML-IT2-FELM that is inspired by kernel learning and called KML-IT2-FELM is also implemented. The results from the recognition of walking activities and gait events achieved an average accuracy of 99.98% and 99.84% with a decision time of 290.4ms and 105ms, respectively, by the ML-IT2-FELM, while the KML-IT2-FELM achieved an average accuracy of 99.98% and 99.93% with a decision time of 191.9ms and 94ms. The experiments demonstrate that the ML-IT2-FELM is not only an effective Fuzzy Logic-based approach in the presence of sensor noise, but also a fast extreme learning machine for the recognition of different walking activities.

Keywords: Multilayer Neural Networks (ML-NNs), Fuzzy Autoencoders (FAEs), Interval Type-2 Fuzzy Logic System (IT2 FLSs), Wearable Sensors, Kernel-based ELM, Direct-defuzzification method, Extreme Learning Machine.

1. Introduction

Recognition of human activities has become very popular during the last decade [1], particularly in fields such as bio-medicine, elderly care, sports injury detection, entertainment, military devices, pattern recognition and soft Robotics, innumerable applications can be found [1–8]. This is mainly due to new advances in the area of machine learning and artificial intelligence, as well as the development of new technology and mobile devices [2]. The recognition of human activities has been mainly approached in two different ways, namely, using wearable and external sensors [8]. By using wearable sensors, electronic devices are attached to the main user, while for external

sensors, devices are fixed in predetermined points of interest making the inference engine to be fully dependant on the voluntary interaction of the users with sensors [8]. In healthcare, the use of wearable sensors for the recognition and monitoring of human daily activities have been mainly applied to assist disabled and elderly people [9]. Particularly to detect abnormal situations or prevent unpredictable events such as falls [9, 10]. Within this context, human locomotion activities such as walking, running, standing, laying, walking upstairs and walking downstairs usually involve complex movements that are commonly affected in disabled and elderly people [11, 12]. Although the use of numerous wearable sensors might improve the performance of a recognition algorithm, the design of reliable and robust machine learning methodologies for perception of human walking movements still represent a challenge [13, 14].

In this work, a new Multilayer Interval Type-2 Fuzzy Extreme Learning Machine (ML-IT2-FELM) for the recognition of human walking activities and gait events using wearable sensors is presented. The ML-IT2-FELM is a hierarchical computational architecture that involves two main components. First, unsupervised IT2 fuzzy feature encoding is achieved by stacking a predefined number of FAEs, where the basic building block is based on the functional equivalence between the IT2-RBFNN and a class of IT2 FLSs [15]. Each FAE can be viewed as an independent/separate module whose output layer is a Nie-Tan direct defuzzification method [16]. Finally, the encoded features are classified by using an IT2-FELM with a Karnik-Mendel type-reduction algorithm as its output layer [17]. To study the efficiency of the proposed ML-IT2-FELM with respect to other ML fuzzy neural structures, a kernel-based ML-IT2-FELM that is called KML-IT2-FELM and that is inspired by kernel learning is suggested. Compared to other existing ML networks, both the ML-IT2-FELM and the KML-IT2-FELM do not employ input weights, hence a random projection of inputs and bias of every FAE is not required. In other words, in order to achieve an optimal generalisation performance no random projection mechanism is needed [18]. In this sense, the KML-IT2-FELM eliminates the manual tuning on the number of hidden neurons in every FAE, and exact inverse solution of output weights is guaranteed under invertible kernel matrix [18]. To validate the performance of the ML-IT2-FELM and the KML-IT2-FELM, real data collected from three inertial measurement units (IMUs) attached to the thigh, shanks and foot of a group of twelve people is used to carry out two different experiments. The first experiment consists of the recognition of three different walking activities, i.e. a) Level-Ground Walking (LGW), b) Ramp Ascent (RA) and Ramp Descent (RD), while the second experiment involves the recognition of eight events that describe a phase of stance and swing during the gait cycle. Such periods segment the gait cycle into eight main events, namely, 1) initial contact, loading response, 3) mid stance, 4) terminal stance, 5) pre-swing, 6) initial swing, 7) mid-swing and 8) terminal swing.

The rest of this paper is organised as follows. Section 2 reviews related work, while in section 3, preprocessing of the IMU data set and a background about multilayer ELM and IT2 Fuzzy Extreme Learning Machine (FELM) for classification is provided. Section 4 resents the proposed ML-IT2-FELM and KML-IT2-FELM. In section 5, the performance of the proposed ML methodologies is studied and compared to other existing strategies. Finally, conclusions are drawn in section 6.

2. Related Work

A large number of existing human activity recognition techniques using machine learning can be found in literature [1, 2, 4–8, 10, 13, 14, 19, 20]. In particular, in smart health applications, wearable sensors placed at different body positions represent a realistic mechanism to the processing of locomotion activities, especially to provide assistance and infer human activities during the rehabilitation of elderly people usually from work-related musculoskeletal disorders [21, 22]. Generally, the analysis of movement data involves an approach that first segments the time series data

into contiguous frames or sliding windows. From each window or frame a set of statistical features or from frequency domain is extracted [22]. This feature extraction process has an influence in the recognition algorithm that usually reduces the associated computational load. More recently, the implementation of deep learning strategies for the recognition of human activities has demonstrated a significant improvement in model accuracy while tackling different challenging problems that usually result from multimodal and high dimensional sensor data. This is mainly due to the ability of deep structures to perform automatic feature learning [22–26]. In [22], authors explore three state-of-art deep learning approaches, namely, deep feed-forward networks (DNN), convolutional and recurrent neural network for human activity recognition using wearable sensors. First, deep feed-forward networks with up to five hidden layers were implemented. In [22], the DNN was used as a mechanism to represent a sequence of non-linear transformations to the input data of the network, where each hidden layer contains the same number of nodes and corresponds to a linear transformation based on the rectified linear (ReLU) activation function. The second approach aimed at finding a degree of locality in the patterns matched to the input data while providing a translational invariance of each pattern of movement data. The last approach exploits the ability of recurrent neural networks to find temporal dependencies within movement data. Overall, such approaches achieved an average accuracy of approximately 90%. In [25], authors proposed a Deep Belief Network (DBN) that is composed of multiple Continuous Autoencoders (CAEs) to classify nineteen human activities. Data was collected using body-worn inertial sensors, magnetic sensors and accelerometers. The architecture of each CAE is based on an stochastic neural network. In order to reduce the number of initial features and shorten the training time, a time and frequency domain feature extract (TFFE) and a fast stochastic gradient descent algorithm method were implemented respectively. Based on the results presented in [25], the proposed DBN achieved an overall accuracy of 99.3%. However, deep learning approaches usually are trained by slow learning speeds. Compared to deep learning, Multilayer Extreme Learning Machine (ML-ELM) has demonstrated to be a faster learning method aiming to solve the heavy computational burden that results from learning methods such as gradient descent and quadratic programming. ML-ELM implementations have also shown superb efficiency in a number of different applications such as image processing [27], pattern recognition [18, 28, 29], complex systems modelling [30] and human-centered computing [7]. For example, in [7], a novel ML-ELM that stacks a number of ELM autoencoders and incorporates a kernel-risk-sensitive-loss measure was developed to identify the motion sequence of body based on the skeleton data. The proposed ML methodology that is called S-ELM-KRSL has multiple hidden layers where each one is an autoencoder that performs unsupervised learning and feature representation. From the experimental results presented in [7], the proposed S-ELM-KRSL achieves a higher accuracy compared to traditional methods while providing a robust performance in the presence of noisy signals and outliers.

In this paper, a Multilayer Interval Type-2 Fuzzy Extreme Learning Machine (ML-IT2-FELM) for the recognition of human walking activities is developed. The proposed ML-IT2-FELM is a hierarchical ML network that consist of two main components. First, multiple IT2 Fuzzy Autoenconders (FAEs) are stacked for unsupervised learning and feature extraction. Secondly, an Interval Type-2 Fuzzy ELM whose neural structure is based on the functional equivalence between the IT2-RBFNN and IT2 FLSs is applied to classify the extracted features by the first component. Compared to traditional approaches and deep learning structures, the proposed ML-IT2-FELM incorporates on its neural structure the ability of IT2 FLSs to deal with complex signals and function approximation in the presence of noise on the one hand.

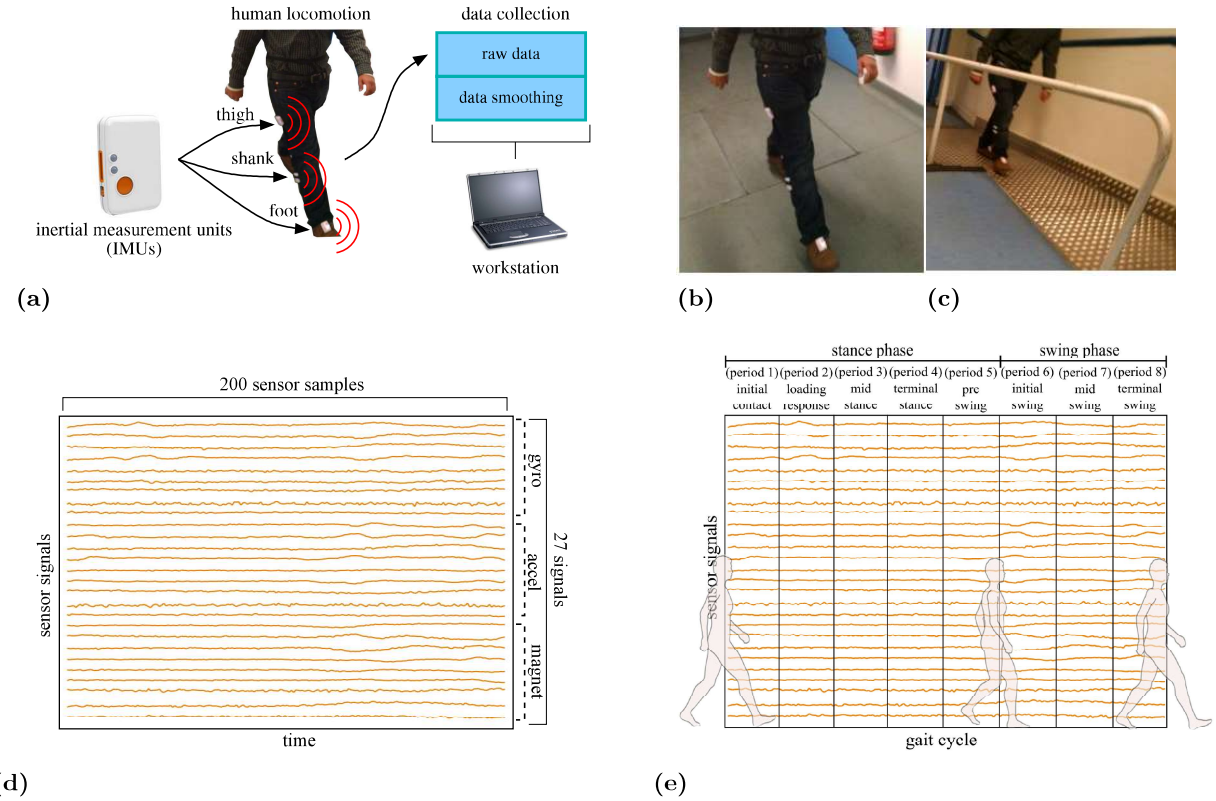

Figure 1: Sensor signals for recognition and prediction of walking activities and gait periods. (A) Data collection from 9-DoF inertial measurement units (IMU) attached to the thigh, shank and foot. (B) Example of concatenated signals from gyroscope (x,y,z), accelerometer (x,y,z) and magnetometer (x,y,z) for a walking activity. (C) Segmentation of the dataset, into 8 periods, for recognition and prediction of gait periods and phases during a walking activity.

On the other hand, the ML-IT2FELM inherits the ability of ML-ELM to provide a high trade-off between model accuracy and a low computational load. A detailed description of the ML-IT2-FELM is provided in the next sections.

3. Preliminaries

This section provides a description about the processing of the IMU data set as well as background that is needed to describe the proposed ML-IT2-FELM and the KML-IT2-FELM is reviewed.

3.1. Signal Processing and Data Collection

Angular velocity signals were employed from three inertial measurement units (IMU, see Figure 1a), worn by 12 healthy human participants. Anthropometric data from participants are as follows: ages between 24 and 34 years old, heights between 1.70 m and 1.82 m, and weights between 75.5 kg and 88 kg. Participants were asked to walk at their self-selected speed and perform ten repetitions of three walking activities: level-ground walking, ramp ascent and ramp descent. Level-ground walking was performed on a flat cement surface. A metallic ramp, with a slope of 8.5 deg, was used for ramp ascent and descent (Figures 1b,c). Sensor signals were systematically collected and filtered with a cut-off frequency of 10 Hz. For this process, a number of three IMUs (Shimmer Inc.) attached to the thigh, shank and foot of each participant were employed.

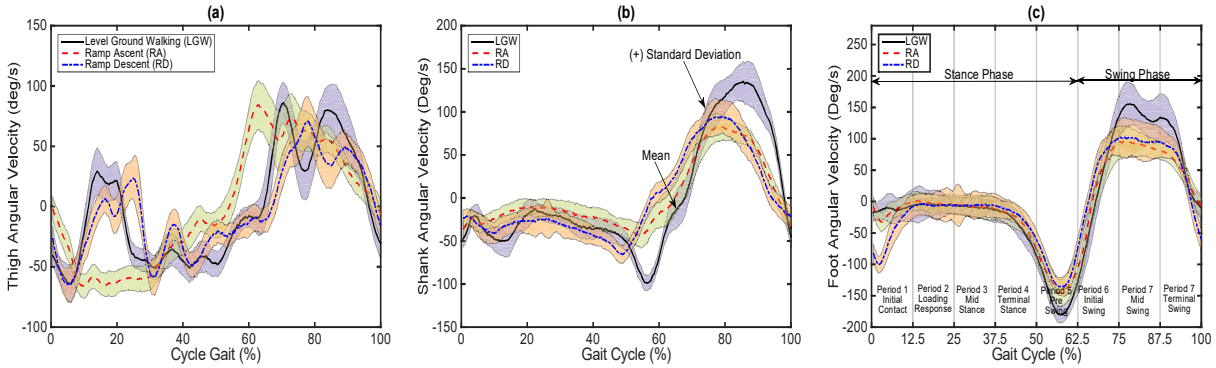

Figure 2: Example angular velocity data collected from three locomotion modes: Level-Ground Walking (LGW), Ramp Ascent (RA), Ramp Descent (RD) using three inertial measurement units (IMUs) attached to the (a) thigh, (b) shank and (c) foot of healhty humans. The data were collected using three inertial measurement units attached to (A) the thigh, (B) shank and (c) foot of healthy participants. Plot (c) shows an example of the gait cycle which is segmented into eight periods, i.e. 1) initial contact, 2) loading response, 3) mid stance, 4) terminal stance, 5) pre-swing, 6) initial swing, 7) mid swing and 8) terminal swing. Such gait events are processed by the proposed Multilayer Fuzzy Extreme Learning Machine models (ML-FELM) for the recognition of human locomotion into three different walking activities.

For each IMU, angular velocity signals in x-y and-z axes, were sampled at 100 Hz. These signals were concatenated to form an initial dataset (as detailed in Fig. 1(d)), composed of 9 signals (1 signal $\times$ 3 axes $\times$ 3 sensors) and 200 sensor samples, from each activity performed by participants. Thus, a dataset of 3×1024 samples was created, where each input is a dimensional array of 9×200 values. Datasets from 8 and 4 participants were used to train and test the proposed Multi layer strategies, respectively. Fig. 1(d) shows an example of the signals collected from the wearable sensors during a walking activity. In addition, two foot pressure-insole sensors were used to detect the beginning and end of each gait cycle. Fig. 1(e) presents the segmentation of the gait cycle into stance phase, swing phase and eight periods (initial contact, loading response, mid-stance, terminal stance, pre-swing, initial swing, mid-swing, terminal swing). This segmentation allows the proposed strategy to recognise and predict the state of the human body during a walking activity. Hence, 25 signal readings (or 15% out of the total gait cycle) were collected to describe each gait event as illustrated in Fig. 2.

3.2. Multilayer Extreme Learning Machine (ML-ELM)

Multilayer Extreme Learning Machine (ML-ELM) was suggested in [27] as a fast and accurate alternative to iterative Back Propagation (BP) learning algorithms that are frequently employed to train Multilayer Neural Networks (ML-NNs) [31–33]. The main advantage of an ML-ELM is the integration of a single learning mechanism that involves several layers for representational learning, followed by a final layer of ELM classification [27].

The basic idea behind ML-ELM is an ELM-based Autoencoder (AE, See Fig. 3) that is stacked to build a multilayer structure (deep structure) while performing layer by layer unsupervised learning for feature representation where fine iterative tuning is not required [34]. As detailed in [18, 28], an ML-ELM is a neural structure that consists of a number of L hidden layers, where for a given input $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_N^{(i)}]$, $k = 1, \ldots, N$, the ith data transformation $\Gamma^{(i)} = [\gamma_1^{(i)}, \ldots, \gamma_N^{(i)}]$ is computed using Eq. (1), and $\gamma^{(i)}$ is the transformation vector that corresponds to the input vector $\mathbf{x}_k^{(i)}$.

$$\mathbf{H}^{(i)}\Gamma^{(i)} = \mathbf{X}^{(i)} \tag{1}$$

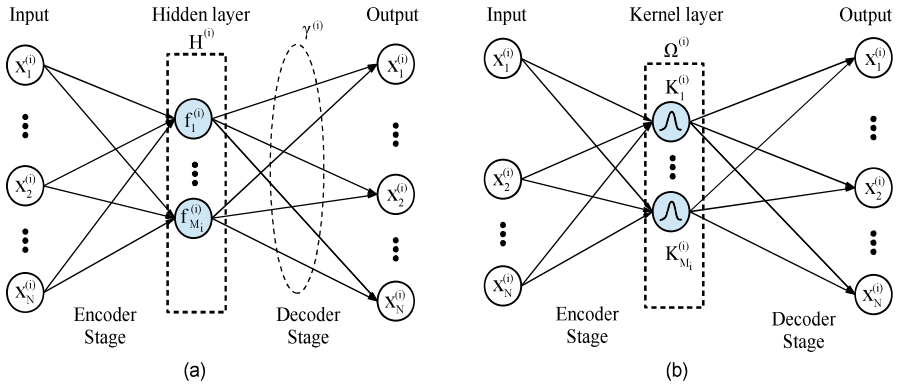

Fig. 3. Architecture of the ith ELM-AE used as the basic building block of a ML-ELM (Taken from [28]).

in which, $\mathbf{H}^{(i)}$ is the output matrix of the ith hidden layer w.r.t the input $\mathbf{X}^{(i)}$. As illustrated in Fig. 1, data transformation is achieved by projecting $\mathbf{X}^{(i)}$ along the decoder stage weights at each ELM-AE. That means, at the encoder stage, each ELM-AE generates a number of orthogonal random parameters, e.g. random input weights and random biases in hidden nodes for additive nodes, or random centers and impact factors for Radial Basis Functions (RBFs) [34, 35]. Thus, orthogonal random hidden parameters of linear and non-linear ELM-AE are computed using Eq. 2.

$$\mathbf{h}(\mathbf{x}_k) = g(\mathbf{x}_k\mathbf{A} + \mathbf{b}) = [h_1(\mathbf{x}_k), \ldots, h_{M_i}(\mathbf{x}_k)] \tag{2}$$

in which, $\mathbf{H}^{(i)} = [\mathbf{h}(\mathbf{x}_1), \ldots, \mathbf{h}(\mathbf{x}_N)]^T$, $\mathbf{A}^T\mathbf{A} = \mathbf{I}$ and $\mathbf{b}^T\mathbf{b} = 1$. Hence, the ith transformation term $\Gamma^{(i)}$ is calculated as:

$$\Gamma^{(i)} = (\mathbf{H}^{(i)})^T \left(\frac{\mathbf{I}}{C} + \mathbf{H}^{(i)}(\mathbf{H}^{(i)})^T \right)^{-1} \mathbf{X}^{(i)} \tag{3}$$

Thus, the final representation $\mathbf{X}^{\text{final}}$ is obtained by multipliying $\mathbf{X}^{(i)}$ with $\Gamma^{(i)}$

$$\mathbf{X}^{\text{final}} = g(\mathbf{X}^{(i)}(\Gamma^{(i)})) \tag{4}$$

As suggested in [27], if layer $M_i = M_{i+1}$, then g can be chosen as a linear function, otherwise g is chosen as nonlinear piecewise function such as RBFs or sigmoids. In [28], $\mathbf{X}^{\text{final}}$ was used as the hidden layer output to compute the output weight vector β as [28].

$$\mathbf{X}^{\text{final}}\beta = \mathrm{T} \tag{5}$$

by adding a regularisation factor C and using the pseudoinverse of the final transformation, $\mathbf{X}^{\text{final}}$ as the firing strength matrix, the term β is computed as shown in Eq. 6.

$$\beta = \mathbf{X}^{\text{final}} \left(\frac{1}{C} + \mathbf{X}(\mathbf{X}^{\text{final}}) \right)^{-1} \mathbf{T} \tag{6}$$

Unlike the hierarchical ML ELM suggested in [27], the methodology reviewed in this section directly uses the final data representation $\mathbf{X}^{\text{final}}$ as hidden layer to calculate the weight vector β.

3.3. Kernel-based Multilayer Extreme Learning Machine (ML-KELM)

Compared to ML-ELM [28], an ML-KELM is a fast multilayer neural structure inspired by kernel learning whose main advantages lie not only on the elimination of manual tuning on the number of hidden nodes in every layer but also to improve the suboptimal model generalisation that usually results from a random projection of input weights and bias in every layer of an ML-ELM [18]. As pointed out in [18], an ML-KELM also provides an exact inverse solution for output weights under invertible kernel matrix. As detailed in Fig. 3, an ML-KELM learns feature/data representation by stacking a number of $'L'$ KELM Autoencoders (AE). At the hidden layer of each AE, a kernel function $K^{(i)} = exp(- \parallel x_p - x_j \parallel /2\sigma_j^2)$ is employed (Radial Basis Function was suggested in [18]). Thus, each input vector $\mathbf{x}_p$ is mapped into a kernel matrix Ω^i where the term $\tilde{\Gamma}$ is used to represent the ith transformation matrix in the KELM-AE, which can be computed as follows:

$$\Omega^{(i)}\tilde{\Gamma}^{(i)} = \mathbf{X}^{(i)} \tag{7}$$

By applying the Mercer's conditions on ELM, the kernel matrix Ω in the hidden layer of each KELM-AE can be written as

$$\Omega^{(i)} = \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_{M_i}) \end{bmatrix} \tag{8}$$

where $\Omega_{j,l} = K(\mathbf{x}_j, \mathbf{x}_l),\ j, l = 1, \ldots, M_i$.

$$\tilde{\Gamma}^{(i)} = \left(\frac{\mathbf{I}}{\mathbf{C}} + \Omega^{(i)} \right)^{-1} \mathbf{X}^{(i)} \tag{9}$$

such as C is the user-specified term for regularisation, $\tilde{\Gamma}^{(i)} = [\gamma_1^{(i)}, \ldots, \gamma_M^{(i)}]$, where $\gamma^{(i)}$ is the ith transformation vector used for representation learning to the input data $\mathbf{X}^{(i)}$. The final transformation $\mathbf{X}^{(i+1)}$ is obtained using a sigmoid function

$$\mathbf{X}^{(i+1)} = g\left(\mathbf{X}^{(i)} (\tilde{\Gamma}^{(i)})^T \right) \tag{10}$$

As indicated out in [18, 27, 32], if the ith transformation has the same dimension as the $(i+1)th$ layer, the activation function $'g'$ can be chosen as a linear function. $\Omega^{(i)}$ is a matrix of semidefinite kernels K where a Reproducing Kernel Hilbert Space (RHKS) that coincides with an N-dimensional space is spanned by a class of finite number of continuous positive semidefinite kernels such as the Radial Basis Functions (RBFs) [36]. Within this context, a KELM is a kernel method that can be viewed as a case of Single Layer Feedforfward Networks (SLFN) with additive neurons [37].

3.4. Interval Type-2 Fuzzy Extreme Learning Machine for Classification (IT2-FELM)

Most of the applications and design of hybrid neural network systems whose structure is based on the theory of Interval Type-2 Fuzzy Logic (IT2 FL) and Extreme Learning Machine (ELM) have been concentrated on solving regression problems. Such systems usually employed Multiple-Input-Single-Output (MISO) neural structures with a Karnik-Mendel type-reduction layer [30, 38, 39]. In this paper, the final layer of the ML-IT2-FELM is an IT2-FELM whose main inference engine is based on the model of a Multi-Input-Multi-Output (MIMO) IT2-RBFNN [28]. Thus, the proposed MIMO IT2-RBFNN is applied to the classification of the set of features created in the final transformation [40]. Both models, the RBFNN and the IT2-RBFNN are a class of MIMO ELMs that were initially suggested for function approximation [41]. Based on the functional equivalence between the IT2-RBFNN and IT2 FLSs [40, 42], an ELM can be constructed as an

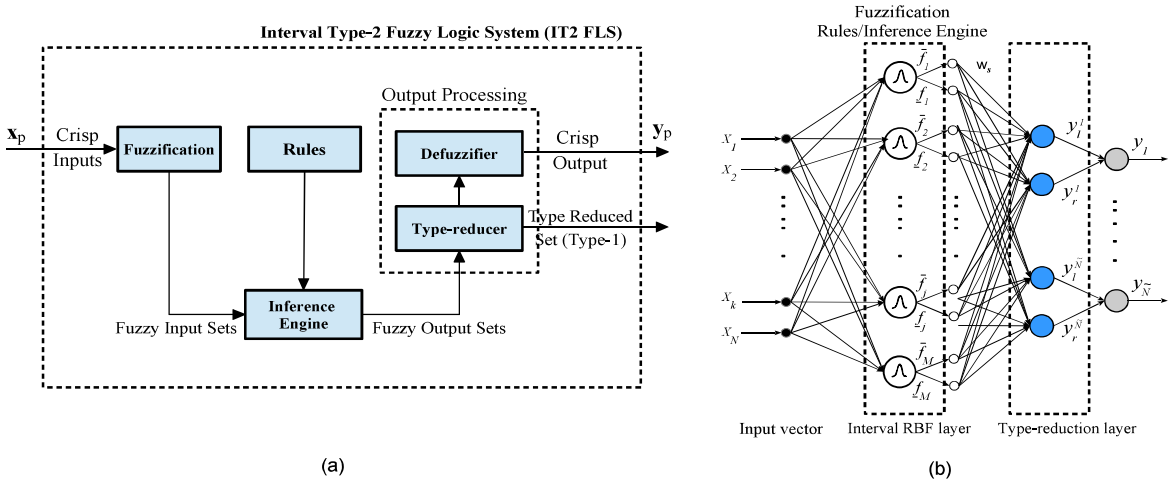

Fig. 4. Architecture of an (a) MIMO IT2 FLS and an (b) MIMO IT2 ELM based on the functional equivalence between the IT2 RBFNN and IT2 FLSs for classification.

MIMO IT2 FLS if its associated inference mechanism is interpreted as an adaptive filter which resembles an additive combination of the MFs (firing strengths). As illustrated in Fig. 4, given an input vector $\mathbf{x}_p = [x_1, \ldots, x_N]^T$, each corresponding fuzzy rule R^j in an MIMO IT2 is described by a multi-variable Gaussian MF $\mu_{R^i}(\mathbf{x}_p, y_p^s) = \mu_{R^j}[x_1, \ldots, x_n, y_p^s],\ s = 1, \ldots, \tilde{N}$, where the input vector $\mathbf{x}_p \in X_1 \times \ldots X_N$ and the implication engine is defined as:

$$\mu_{R^j}(\mathbf{x}_p, y_p^s) = \mu_{A^j \to G^j} = \left[T_{k_1}^N \mu_{F_k^j}(x_k) \star \mu_{G^j(y)}\right] = [\underline{f}_j(\vec{x}_p), \bar{f}_j(\vec{x}_p)] \tag{11}$$

where $\star$ is the minimum $t-norm$ that represents the shortest Euclidean distance to the input vector $\mathbf{x}_p$. $[\underline{f}_j(\vec{x}_p), \bar{f}_j(\vec{x}_p)]$ is the lower and upper membership function (LMF, UMF) respectively. In this work, each MF in the fuzzy rule of the MIMO IT2-RBFNN is an interval Gaussian MF with an uncertain width $\sigma_j = [\sigma_j^1, \sigma_j^2]$ and fixed center (mean) μ_{kj} as shown in Fig. 5:

$$F^j := \begin{cases} F^j = [\underline{f}_j(\mathbf{x}_p), \bar{f}_j(\mathbf{x}_p)] \\ \underline{f}_j(\mathbf{x}_p) = \exp\left[-\sum_{k=1}^{N}\left(\frac{x_k - \mu_{kj}}{2\sigma_j^2}\right)^2\right] \\ \bar{f}_j(\mathbf{x}_p) = \exp\left[-\sum_{k=1}^{N}\left(\frac{x_k - \mu_{kj}}{2\sigma_j^1}\right)^2\right] \end{cases} \tag{12}$$

The jth fuzzy rule in an MIMO IT2-RBFNN is written as

$$\tilde{R}^j : \text{IF } x_1 \text{ is } F_1^j \text{ and} \ldots \text{IF } x_k \text{ is } F_s^j \text{ and} \ldots \text{IF } x_N \text{ is } F_N^j \text{ THEN } y \text{ is } w_{jk};\ j = 1, \ldots, M \tag{13}$$

For an IT2-RBFNN of Mamdani type (also known Zadeh type), the weight vector (consequent) w_s is a vector of single crisp values, while for a TSK model, each term $w_{js} = c_0^s + c_1^s x_1 + c_2^s x_2 + \ldots + c_N^s x_N$. Based on the functional equivalence between the RBFNN and IT2 FLSs [40, 43], in this paper, for each output y_s, the MIMO IT2-RBFNN is a FLS having a center-of-sets type reduction, product inference rule and a singleton output space. The type-reduced set (y_l, y_r) is obtained by using a Karnik-Mendel algorithm [17].

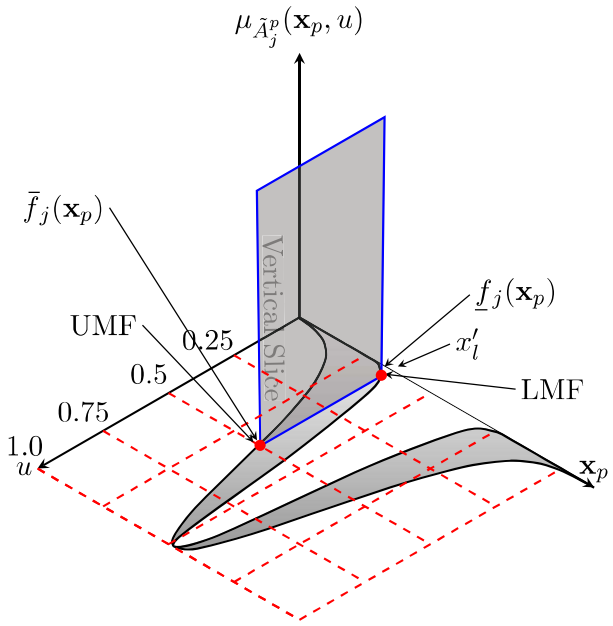

Figure 5: Singleton fuzzification and interval secondary MF that is activated when $\mathbf{x}_p = x_l'$ for the jth receptive unit of the IT2-RBFNN (Taken from [30]).

According to Fig. 4, if w_{js} is a crisp value and the inference engine for the IT2-RBFNN is either of Mamdani or TSK type, the matrix representation of the sth output in the MIMO IT2-RBFNN can be written as [17, 44]:

$$y_s = \frac{1}{2}\left(\mathbf{Y}_l^s + \mathbf{Y}_r^s\right)\mathbf{w}_s^T,\ \ s = 1, \ldots, \tilde{N} \tag{14}$$

in which $y_l^s = \mathbf{Y}_l^s \mathbf{w}_s^T$ and $y_r^s = \mathbf{Y}_r^s \mathbf{w}_s^T$ and

$$\mathbf{Y}_l^s = \frac{\bar{\mathbf{f}}^T Q_s^T E_{1s}^T E_{1j} Q_s + \underline{\mathbf{f}}^T Q_s^T E_{2s}^T E_{2s} Q_s}{r_l^T Q_s \bar{\mathbf{f}} + \hat{s}_{ls}^T Q_s \underline{\mathbf{f}}} \tag{15}$$

In which $\mathbf{w}_s = (w_{1s}, \ldots, w_{Ms})$ is the set of original rule-ordered consequent weights, and $\mathbf{Y}_l^s = (\psi_{ls,1}, \ldots, \psi_{ls,M})$, and the terms E_{1s}, E_{2s}, r_{ls} and $\hat{s}_{ls}$ are defined as:

$$\begin{aligned}
E_{1s} &= (e_{1s}|e_{2s}|\ldots|e_{Ls}|\mathbf{0}|\ldots|\mathbf{0})^T \ \ L_s \times M \\
E_{2s} &= \left(\mathbf{0}|\ldots|\mathbf{0}|\xi_1^s|\xi_2^s|\ldots|\xi_{M-L_s}^s\right)^T \ \ (M - L_s) \times 1 \\
r_{ls} &\equiv (\underbrace{1, 1, \ldots, 1}_{L}, 0, \ldots, \ldots, 0)^T \ \ M \times 1 \\
\hat{s}_{ls} &\equiv (0, \ldots, \ldots, 0 \overbrace{1, 1, \ldots, 1}^{M-L_s})^T \ \ M \times 1
\end{aligned}$$

in which L_s is the switching point that corresponds to the sth output, $e_m \in R_s^L$ $(m = 1, \ldots, L_s)$ and $\xi_m \in R^{M-L_s}$, $m = 1, \ldots, M - L_s$ as the elementary vectors where all the elements are zero except the mth one that is equal to 1.

$$\mathbf{Y}_r^s = \frac{\underline{\mathbf{f}}^T Q_s^T E_{3s}^T E_{3s} Q_s + \bar{\mathbf{f}}^T Q_s^T E_{4s}^T E_{4s} Q_s}{r_{rs}^T Q_s \underline{\mathbf{f}} + \hat{s}_{ls}^T Q_s \bar{\mathbf{f}}} \tag{16}$$

where $\mathbf{Y}_r^s = (\psi_{rs,1}, \ldots, \psi_{rj,M})$

$$
\begin{aligned}
E_{3s} &= (e_{1s}|e_{2s}|\ldots|e_{Rs}|\mathbf{0}|\ldots|\mathbf{0})^T \;\; R_s \times M \\
E_{4s} &= (\mathbf{0}|\ldots|\mathbf{0}|\xi_{1s}|\xi_{2s}|\ldots|\xi_{M-R_s})^T \;\; (M-R_s)\times 1 \\
r_{rs} &\equiv (\underbrace{1,1,\ldots,1}_{R_s},0,\ldots,\ldots,0)^T \; M\times 1 \\
\hat{s}_{rs} &\equiv (0,\ldots,\ldots,0\overbrace{1,1,\ldots,1}^{M-R_s})^T \; M\times 1
\end{aligned}
$$

with $e_m \in R^{Rs}$ $(m = 1, \ldots, R_s)$ and $\xi_m \in R^{M-R_s}$, $s = 1, \ldots, M - R_s$ as the elementary vectors where all the elements are zero except the mth one that is equal to 1 [45]. $\underline{\mathbf{f}} = (\underline{f}_1, \ldots, \underline{f}_M)^T$, $\bar{\mathbf{f}} = \left(\bar{f}_1, \ldots, \bar{f}_M,\right)^T$. By using Karnik-Mendel algorithms [17], the reordered consequent weight $\tilde{\mathbf{w}}_s$ that results from the permutation process to finding the switching points L and R is obtained as follows [45]:

$$\tilde{\mathbf{w}}_s = Q_s \mathbf{w}_s^T, \quad Q_s \in R^{M\times M} \tag{17}$$

in which Q_s is the corresponding permutation matrix [45]. Hence, the pth MIMO IT2-RBFNN output is the vector of $\tilde{N}$ outputs $\mathbf{Y}_p = \left[y_1, \ldots, y_{\tilde{N}}\right]^T$. As pointed out in , in order to consider a of system IT2, only one out of its components must be of IT2. Within this context, the Thereby, in order to find the parameters of the MIMO IT2-RBFNN, ELM is systematically called in two different steps in order to update the consequent weights in the IT2-RBFNN output layer [30, 38]. At first step [38], the optimal initial values for the consequents are obtained by approximating the reduced set for the sth output $[y_l^s, y_r^s]$ as:

$$y_{l,1}^s = \frac{\sum_{j=1}^M \underline{f}_j w_{js}}{\sum_{j=1}^M \underline{f}_j} = \sum_{j=1}^M \underline{f}'_j w_{js};\;\; \underline{f}'_j = \frac{\underline{f}_j}{\sum_{j=1}^M \underline{f}_j} \tag{18}$$

$$y_{r,1}^j = \frac{\sum_{j=1}^M \bar{f}_j w_{js}}{\sum_{j=1}^M \bar{f}_j} = \sum_{j=1}^M \bar{f}'_j w_{js};\;\; \bar{f}'_j = \frac{\bar{f}_j}{\sum_{j=1}^M \bar{f}_j} \tag{19}$$

where the weight vector $\mathbf{w}_s = [w_{1s}, \ldots, w_{Ms}]^T$ and the weight matrix is defined as $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}]$. By using Eq. (15) and (16), the following linear system can be written for a number of P patterns [46]:

$$\mathbf{T} = \mathbf{\Phi_A}(\mathbf{x})\mathbf{W},\;\; \mathbf{W} \in \mathbf{R}^{M\times\tilde{N}} \tag{20}$$

in which, $\mathbf{T} = [t_1, \ldots, t_P]^T$, is the desired output vector, $p = 1, \ldots.P$ and each $t_p = [t_{1j}, \ldots, t_{Mp}]^T$. For an IT2-RBFNN with a TSK (Mamdani) fuzzy rule structure, matrix Φ_A can be written as.

$$\mathbf{\Phi_A}(\mathbf{x}) = \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_P \end{pmatrix} \in R^{P\times(M\times N)} \tag{21}$$

From Eq. (18) and (19) it follows for a TSK implication:

$$\Phi_p \mathbf{W}_1 = \frac{1}{2}\sum_{j=1}^M (\bar{f}'_j + \underline{f}'_j)\left[\sum_{k=1}^N c_k^{j1} x_k, \ldots, \sum_{k=1}^N c_k^{j\tilde{N}} x_k\right];\;\; s = 1, \ldots, \tilde{N} \tag{22}$$

For an IT2-RBFNN of Mamdani type, the second addition term in Eq. (25) is a single crisp value w_{js}. Therefore, the solution to the linear system described in Eq. (20) is calculated as follows:

$$\mathbf{W}_1 = \Phi_{\mathbf{A}}(\mathbf{x})^{\dagger}\mathbf{T} \tag{23}$$

where $\mathbf{W}_1$ is the optimal initial value for the consequent matrix $\mathbf{W}$ and $\Phi_{\mathbf{A}}(\mathbf{x})^{\dagger}$ is the Moore-Penrose generalised inverse of $\Phi_{\mathbf{A}}(\mathbf{x})$. Secondly, the final optimisation of $\mathbf{W}$ consists of implementing $\tilde{N}$ times the Karnik-Mendel algorithm. That is, each column vector in the matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}]^T$ is used to form a linear system given by

$$\mathbf{t}_s = \Phi^s_{\mathbf{B}}(\mathbf{x})\mathbf{w}_s, \mathbf{w}_s \in \mathbf{R}^M \tag{24}$$

Therefore, From Eq. (15) and (16) the terms $\mathbf{Y}^s_l$ and $\mathbf{Y}^s_r$ are used to calculate each weight vector $\mathbf{w}_s$ where each $\Phi^s_{\mathbf{B}}(\mathbf{x})$ becomes

$$\Phi^s_{\mathbf{B}}(\mathbf{x}) = \begin{pmatrix} \tilde{\Phi}_1 \\ \tilde{\Phi}_2 \\ \vdots \\ \tilde{\Phi}_P \end{pmatrix} \in R^{P\times(M\times N)} \tag{25}$$

Such that

$$\tilde{\Phi}_p\mathbf{w}_s = \sum_{i=1}^{M} h^s_{pj}\left(\sum_{k=1}^{N} c^{js}_k x_{kp}\right) \tag{26}$$

In which, $h^s_{pj} = \frac{1}{2}\left(\psi_{ls,j} + \psi_{rs,j}\right)$. In other words, for classification purposes, the matrix representation of an IT2-RBFNN that is based on ELM can be written as:

$$\Phi^1_{\mathbf{B}}\mathbf{V}_1 + \ldots + \Phi^s_{\mathbf{B}}\mathbf{V}_s + \ldots + \Phi^{\tilde{N}}_{\mathbf{B}}\mathbf{V}_{\tilde{N}} = \mathbf{T} \tag{27}$$

$$\Phi^s_{\mathbf{B}}\mathbf{V}_s = \begin{bmatrix} h^s_{11} & \ldots & h^s_{1M} \\ h^s_{21} & \ldots & h^s_{2M} \\ \vdots & \ldots & \vdots \\ h^s_{P1} & \ldots & h^s_{PM} \end{bmatrix} \begin{bmatrix} 0 & \ldots & w_{1s} & \ldots & 0 \\ 0 & \ldots & w_{2s} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & \ldots & w_{Ms} & \ldots & 0 \end{bmatrix}$$

where $\Phi^s_{\mathbf{B}} \in \mathbf{R}^{P\times M}$, $\mathbf{V}_s \in \mathbf{R}^{M\times\tilde{N}}$, where the target matrix $\mathbf{T}$ is defined as follows:

$$\mathbf{T} = \begin{bmatrix} t_{11} & \ldots & t_{1\tilde{N}} \\ t_{21} & \ldots & t_{2\tilde{N}} \\ \vdots & \ldots & \vdots \\ t_{P1} & \ldots & t_{p\tilde{N}} \end{bmatrix}$$

As described in [42], the IT2-RBFNN for classification problems can be viewed as an ensemble of weighted networks that defines the associated uncertainty of each class as a measure for ambiguity as a variation of the output over unlabeled data sharing the same IT2 firing strength.

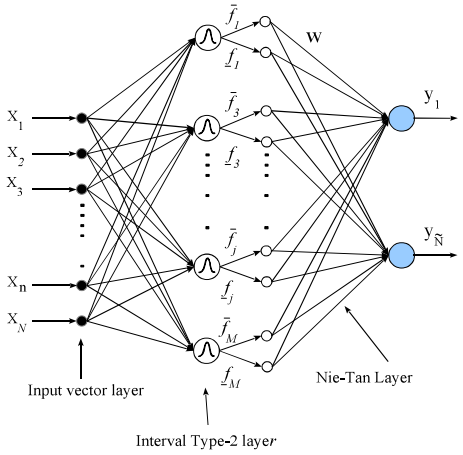

Figure 6: Interval Type-2 Extreme Learning Machine using a NieTan algorithm for classification problems [47].

3.5. Simplified IT2-ELM using the Nie-Tan Method

To reduce the associated computational load that implies the iterative nature of Karnik-Mendel type-reduction methods, a number of close-form algorithms have been suggested [47]. As described in [42], the Nie-Tan is a direct defuzzification process that employs the vertical representation of the Footprint Of Uncertainty (FOU). As illustrated in Fig. 6, an Interval Type-2 Fuzzy Extreme Learning (IT2-FELM) using a Nie-Tan (NT) direct-defuzzification layer does not require a sorting process. The application of a Nie-Tan layer represents a zero Taylor series approximation of Karnik-Mendel+defuzzification method [42]. Moreover, a Nie-Tan operator is equivalent to an exhaustive and accurate type-reduction for both discrete and continous IT2 Fuzzy Sets (FSs) [16]. Therefore, the output of a KIT2-FELM with a NT layer and with a variable width $[\sigma_{1j}, \sigma_{2j}]$ and fixed center can be formulated as follows:

$$\mathbf{T} = \mathbf{H}_{NT}\mathbf{W} \tag{28}$$

where $\mathbf{H}_{NT}$ is the matrix for IT2 firing strengths, and β is the weight vector in the output layer. Thus, $\mathbf{H}_{NT}$ is defined as:

$$\begin{aligned}&\mathbf{H}_{NT}(\mu_1,\ldots,\mu_M,\sigma_{11},\ldots,\sigma_{1M},\sigma_{21},\ldots,\sigma_{2M},\mathbf{x}_1,\ldots,\mathbf{x}_P)\\ &\qquad=\begin{pmatrix}\varphi_{11}(\mu_1,\sigma_{11},\sigma_{21},\mathbf{x}_1) & \cdots & \varphi_{1M}(\mu_M,\sigma_{1M},\sigma_{2M},\mathbf{x}_1)\\ \vdots & \vdots & \vdots\\ \varphi_{P1}(\mu_1,\sigma_{11},\sigma_{21},\mathbf{x}_P) & \cdots & \varphi_{1M}(\mu_M,\sigma_{1M},\sigma_{2M},\mathbf{x}_{MP})\end{pmatrix}_{P\times M}\end{aligned}$$

in which

$$\varphi(\mu,\sigma_{1j},\sigma_{2j},\mathbf{x}_p) = \frac{\underline{f}_j + \bar{f}_j}{\sum_{j=1}^{M}\underline{f}_j + \sum_{j=1}^{M}\bar{f}_j} \tag{29}$$

where $\mu_j = (\mu_{1j},\ldots,\mu_{Nj})$ and w_s and the target $\mathbf{T}$ are defined as follows:

$$\beta = \begin{pmatrix} w_1^T \\ \vdots \\ w_M^T \end{pmatrix}_{M\times\tilde{N}} \quad \text{and} \quad \mathbf{T} = \begin{pmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_P^T \end{pmatrix}_{P\times\tilde{N}} \tag{30}$$

Therefore, the learning parameter for an IT2-ELM using a Nie-Tan algorithm follows a much simpler methodology.

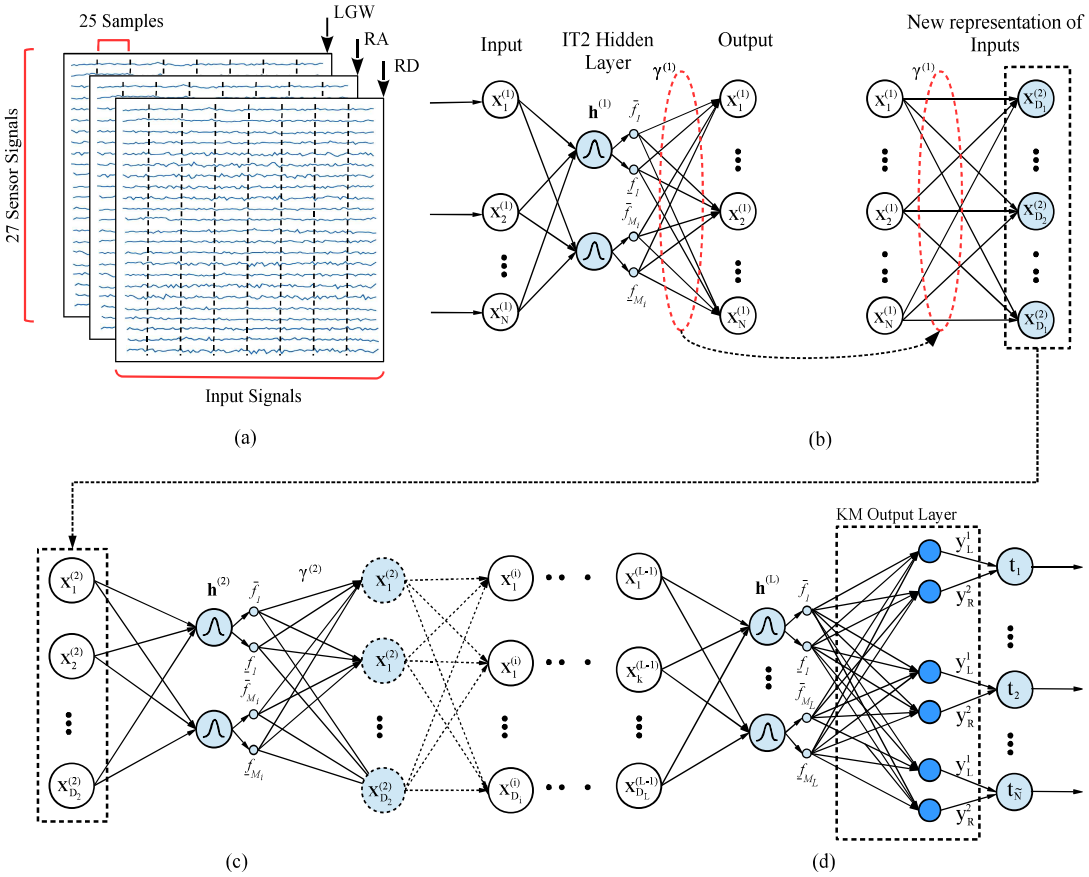

Figure 7: Neural architecture of the ML-IT2FELM, (a) Input data format using 3 wearable sensors, b) IT2 Fuzzy Autoencoder (FAE) based on the NT algorithm, c) L FAEs using the NT algorithm and d) IT2-FELM as using the KM method.

4. Multilayer Interval Type-2 Fuzzy Extreme Learning Machine (ML-IT2-FELM)

The proposed ML-IT2FELM is a multilayer feedforward neural structure that is based on the original ELM algorithm and the theory of Interval Type-2 Fuzzy Sets (IT2 FSs). As illustrated in Fig. 7, the general architecture of the ML-IT2FELM consists of a number of L hidden layers, where the first $L-1$ ones are used to stack a number of Fuzzy Autoencoders (FAEs), followed by a IT2-ELM using a KM type-reduction method. Similar to the hierarchical learning scheme presented in [27], the training of the proposed ML-IT2FELM is a forward methodology that is separated into two phases, in the former phase, a number of $L-1$ FAEs are applied to extract a high feature data representation, while in the latter, this representation is classified using an IT2-ELM with a KM algorithm. On the one hand, to reduce the computational burden that results from the implementation of the iterative nature of KM algorithms, each FAE in the ML-IT2-FELM is an MIMO IT2-ELM that employs a NT algorithm as output layer. On the other hand, to provide a high generalisation performance, an IT2-ELM using a KM is used as a classifier. In this paper, two different learning approaches for the ML-IT2FELM are suggested, i.e. a) a ML architecture based on the hierarchical learning scheme of Stacked Autoencoders (SAE) proposed in [27, 28] in which

the number of hidden untis at each FAE is tuned manually and that we call hierarchical it ML-IT2-FELM for short, and b) an IT2 ML structure based on Kernel extreme learning (kernel-basel ML-IT2-FELM for short) [18].

4.1. Hierarchical ML-IT2-FELM

This methodology involves two hierarchical phases, i.e. first, an initial process for high-level feature extraction is performed by stacking $'L'$ FAEs. At each $'ith'$ hidden layer, an independent IT2-FELM with a NT output layer is implemented. Unlike the ML-ELM suggested in [27] and [28], at layer *'zero'* the input raw data does not need to be transformed into an ELM random feature space. Hence, in the first phase, representation learning of the input raw data is initially performed by constructing the following system:

$$\mathbf{H}_{NT}^{(i)}\tilde{\Gamma}^{(i)} = \mathbf{X}^{(i)},\ i = 1, \ldots, L-1 \tag{31}$$

in which, $\mathbf{H}_{NT}^{(i)} = [h(\mathbf{x}_1), \ldots, h(\mathbf{x}_P)]$ and $\Gamma^{(i)}$ is the ith hidden matrix of IT2 firing strengths and the ith transformation matrix respectively, where $h(\mathbf{x}_p) = [\varphi_{p1}(\mathbf{x}_p), \ldots, \varphi_{pM}(\mathbf{x}_p)]$. $\mathbf{H}_{NT}^{(i)}$ is built using Eq. (19) which results from applying a NT algorithm in the output layer with respect to the input vector $\mathbf{X}^{(i)}$. $[\underline{f}_j(\vec{x}_p), \overline{f}_j(\vec{x}_p)]$ is the jth interval firing strength computed using Eq. (12). Hence, the transformation matrix $\tilde{\Gamma}^{(i)}$ is obtained from Eq. (32) [18]:

$$\tilde{\Gamma}^{(i)} = (\mathbf{H}_{NT}^{(i)})^T \left(\frac{\mathbf{I}}{C} + \mathbf{H}_{NT}^{(i)}(\mathbf{H}_{NT}^{(i)})^T\right)^{-1} \mathbf{X}^{(i)} \tag{32}$$

in which C is the user-specified parameter for regularisation. Thus, $\mathbf{X}^{(i+1)} = g(\mathbf{X}^{(i)}(\Gamma^{(i)})^T)$. In a similar way to [27], $g(\cdot)$ can be any activation function. However, if the dimension of layer (i) and layer $(i+1)$ is the same, it is recommended to chose a linear piecewise function [18]. In the second phase, the final hidden layer L is an IT2-FELM whose input is $\mathbf{X}^{(i+1)}$. As described in the previous section, in order to find the parameters of the IT2-FELM, ELM is systematically called in two different steps in order to update the consequent weights in the IT2-FELM output layer [30, 38].

4.2. Kernel-Based ML-IT2-FELM

The second learning approach for the ML-IT2-FELM is based on Kernel ELM [18] (called KML-IT2-FELM for short). Similar to the KML-ELM, the KML-IT2-FELM eliminates the manual tuning on the selection of the number of hidden nodes at each kernel-based FAE which is defined as

$$\Omega_{NT}^{(i)}\tilde{\Gamma}^{(i)} = \mathbf{X}^{(i)} \tag{33}$$

in which, $K(\mathbf{x}_k, \mathbf{x}_l) = [\underline{f}_j(\vec{x}_p), \overline{f}_j(\vec{x}_p)]$, and the ith matrix transformation $\tilde{\Gamma}^{(i)}$ is calculated as:

$$\tilde{\Gamma}^{(i)} = \left(\frac{\mathbf{I}}{\mathbf{C}} + \Omega_{NT}^{(i)}\right)^{-1} \mathbf{X}^{(i)} \tag{34}$$

In a like manner to the learning method developed for the ML-IT2-FELM, if the dimension of layer (i) and layer $(i+1)$ is the same, it is recommended to chose a linear piecewise function [18]. Thus, the $(i+1)th$ transformation is used as the input of an IT2-FELM where the matrices $\Phi_{\mathbf{A}}$ and $\Phi_{\mathbf{B}}^{s}$ in Eq. (23) and (25) are replaced by the corresponding kernel matrices $\Omega_{\mathbf{A}}$ and $\Omega_{\mathbf{B}}^{s}$ respectively. In other words, the number of hidden units is equal to the number of inputs. Thereby, a KML-IT2-FELM can be viewed as a ML FLS of IT2 if [40, 42]:

1. The input layer of each kernel-based FAE performs singleton fuzzification and the MFs within each rule are chosen as Radial Basis Functions of Interval Gaussian type [41, 43].

2. The number of kernels ($K(\mathbf{x}_j, \mathbf{x}_l),\ j, l = 1, \ldots, N$) is equal to the number of IF-THEN fuzzy rules. The output of each kernel is an interval, i.e. $k(\mathbf{x}_j, \mathbf{x}_l) = [\underline{f}_j(\vec{x}_p), \bar{f}_j(\vec{x}_p)]$

4. The T-norm operator used to compute each rule's firing strength in the hidden layer is multiplication (meet).

5. A type-reduction algorithm such as the KM or a direct defuzzification method is employed to find the FLS output.

Thus, any kind of FLS enhancement might be directly applicable to each hidden layer of the ML-IT2-FELM models because the structure of its fuzzy rule base in going from T1 Fuzzy Sets (FSs) to higher order FSs does not change; it is the way the associated antecedent and consequent parts are modelled [44, 48].

5. Experiments and Evaluation

In this section, the model evaluation of the proposed ML-IT2-FELM structures is conducted on two different experiments, i.e. for the recognition of 1) walking activities and 2) gait events. The former involves the recognition of three walking activities: a) level-ground walking (LGW), b) ramp ascent (RA) and c) ramp descent (RD), while the latter consists of the recognition of gait events which are divided into eight periods: (1) initial contact, (2) loading response, (3) mid stance, (4) terminal stance, (5) pre-swing, (6) initial swing, (7) mid swing and (8) terminal swing. Experiments are carried out using MATLAB 2014b running on a 3.6-GHz i7 CPU with 16-GB RAM. Model performance of the ML-IT2-FELM and KML-IT2-FELM and their associated computational burden are compared to the performance of other existing machine learning approaches such as the ML-ELM [28], a multilayer neural network (ML-RBFNN) based on the RBFNN architecture [42], an ensemble of classifiers [49], a DBN [50], Support Vector Machines (SVMs) [51], and an adaptive Bayesian inference system (BasIS) [11]. For cross validation purposes, the IMU data set is divided into two subsets, i.e. 85% for training (from 8 healthy participants) and 15% (from 4 healthy participants) for testing.

To more completely understand the non-obvious contributions of each suggested ML method, each experiment consists of ten individual cross validations runs, where the samples for training and testing are randomly selected. For determining the sources of variation in measures of model accuracy, a one-way ANOVA method was implemented. In this work, model assumption for ANOVA method is related to the evidence for the variance in model testing accuracy being influenced by each ML learning methodology.

In this paper, signals about angular velocity (Deg/s) are collected from three IMU sensors and used to train the ML-IT2-FELM and the KML-IT2-FELM models. For both ML models, an Interval Type-2 Gaussian function with a fixed mean m_j and variable width $[\sigma_1, \sigma_2]$ is selected. In other words, an IT2 MF is used as the main fuzzifier for the matrix of firing strengths of a ML-IT2-FELM, and for each ith matrix $\Omega^{(i)}$ composed of a set of Gaussian kernel functions in the KML-IT2-FELM models. Finally, the performance of the proposed models is evaluated in the presence of noisy signals, for each experiment, a Gaussian noise with a signal-to-noise ratio of 50dB was randomly added to each sensor signal.

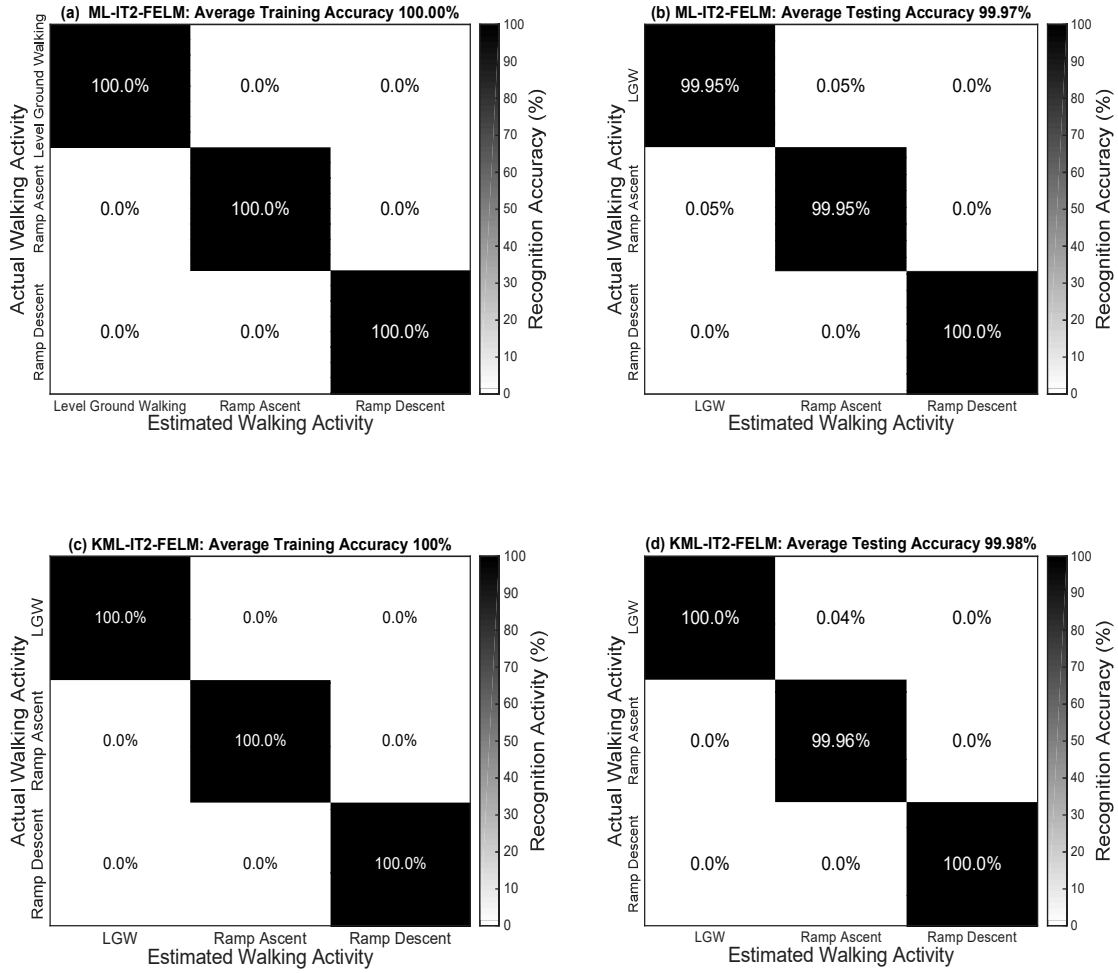

Figure 8: Cross-validation results based on the Confusion matrix and obtained by the ML-IT2-FELM (a,b) and KML-IT2-FELM (c,d) to the recognition of walking activities, i.e. a) Level-Ground Walking (LGW), Ramp Ascent (RA) and Ramp Descent (RD)

5.1. Recognition of Walking Activities

This section presents the model accuracy provided by the ML-IT2-FELM and KML-IT2-FELM to the recognition of walking activities, i.e. 1) Level-Ground Walking (LGW), 2) Ramp Ascent (RA) and 3) Ramp Descent (RD). The experimental setup of ML-IT2-FELM consists of 3 hidden layers, followed by an IT2-FELM classifier, i.e. $L = 4$, set as $4 \times M_i$, $[100 - 500 - 500 - 300 - 3]$ with a regularisation parameter $C_i (i = 1, \ldots, 3)$ defined as $C_i = [0.1, 4.1 \times 10^7, 5 \times 10^8, 55.3]$ while for the KML-IT2-FELM is used as $C_i = [0.1, 4.1 \times 10^2, 10^2, 100]$. Similar to [11], the noise was added to a sensor randomly selected for each decision process performed during the walking activity. Opposite to ML-IT2-FELM, KML-IT2-FELM does not need to manually estimate the number of hidden nodes at each *ith* hidden layer. In Fig. 8, a confusion matrix is used to show the average cross-validation results of ten experiments that correspond the training and testing provided by the ML-IT2-FELM (Fig. 8(a,b)) and the KML-IT2-FELM (Fig. 8(c,d)), where white and black colours represent 0% and 100% recognition accuracy correspondingly. As can observed from Fig. 8(b) and (d), for the recognition of walking activities, both the ML-IT2-FELM and KML-IT2-FELM achieved a testing accuracy of 99.97% and 99.98% respectively.

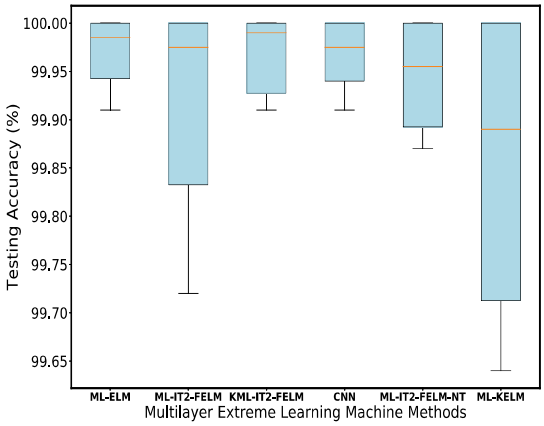

Figure 9: Box-and-whisker diagrams of the average of ten random testing experiments for the recognition of walking activities using six different ML methodologies.

Table 1: One-way ANOVA Method for the recognition of gait events.

Source	SS	df	MS	F-value	P-value
Model Accuracy	0.00943	5	0.01989	2.67	0.0316
Within	0.40246	54	0.00745		
Total	0.50189	59			

The confusion matrix in Fig. 8(c) shows that Level-Ground Walking and Ramp Ascent activities are slightly affected by noisy measurements.

In Fig. 9, a box-and-whisker diagram is presented in order to illustrate the performance distribution of the ML-IT2-FELM and KML-IT2-FELM with respect to other ML neural structures such as: ML-ELM [28], Convolutional Neural Networks (CNN) [5], ML-IT2-FELM-NT and ML-KELM [18]. A ML-IT2-FELM-NT is a ML-IT2-FELM that utilises a Nie-Tan method as a defuzzifier at each hidden layer, including the classification stage. It is apparent in Fig. 9, the distribution of testing performance among models ranges between the values 99.63% and 100%. It can be also observed from Fig. 9, ML-ELM, KML-IT2-FELM and CNN produce the highest accuracy, where their medians of the distributions are within the interval $[99.95\%, 100\%]$. To confirm the model variability presented in Fig. 9, one-way ANOVA results are described in Table 1. As can be observed from Table 1, model assumption for one-way ANOVA is related to the variance affected by each model accuracy (%). A significance value of $\alpha = 0.05$ was employed. From Table 1, it can be concluded that null hypothesis can be rejected as $p - value < \alpha$. This means, not all of population means are equal as illustrated in Fig. 9. To illustrate the level of transparency that can be achieved by the ML-IT2-FELM, the universe of discussion for the MFs that correspond to the first FAE (hidden layer 1) and the IT2-FELM for classification is presented in Fig. 10(a) and (b). To build each universe, the normalised mean value of μkj and $[\sigma_j^1, \sigma_j^2]$ is employed. It can be concluded from Fig. 10, a low level of overlapping is achieved by the first FAE of the ML-IT2-FELM for the feature representation of each sensor signal, while for the IT2-FELM such overlapping is much higher.

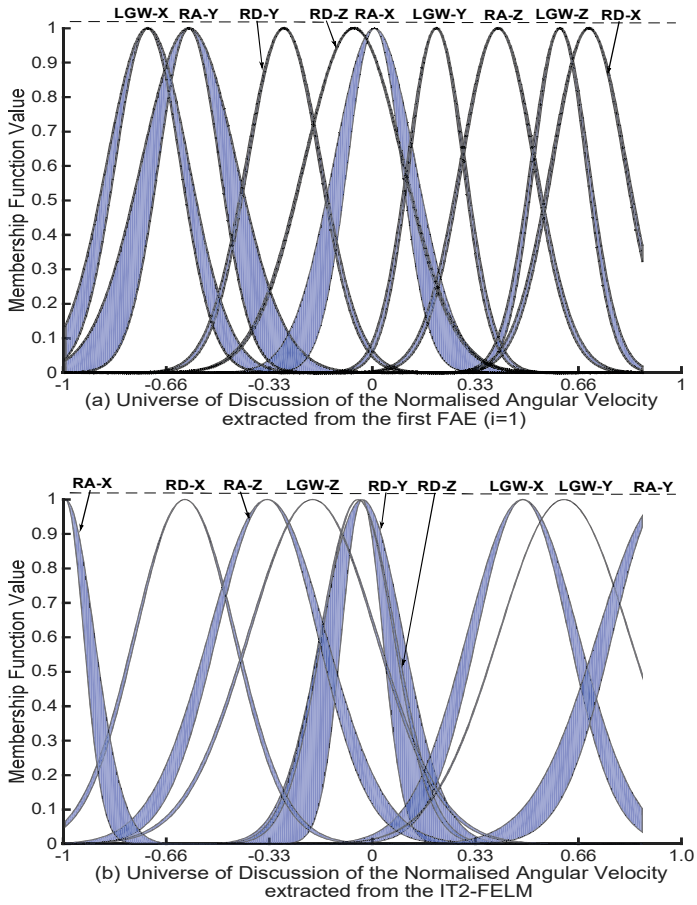

Figure 10: Final Membership Functions (MFs) for the fuzzy sets (FSs): Low Ground Walking for axes X, Y and Z (LGW-X, LGW-Y, LGW-Z), Ramp Ascent for axes X, Y and Z (RA-X, RA-Y, RA-Z) and Ramp Descent for axes X, Y and Z (RD-X, RD-Y, RD-Z) that correspond to the first hidden layer (FAE) of the hierarchical ML-IT2-FELM and the IT2-FELM classifier.

5.2. Recognition of Gait Events

For the classification of gait events, in this experiment the IMU data set was rearranged resulting in an data set of 8192×25 signal recordings. As reported in [52], a set of 25 contiguous signals is suggested to describe each gait event (See Fig. 2(c)). To investigate the optimum number of signal readings (samples or measurements) that is required to recognise a gait event, six experiments using a different average of n_s contiguous samples is implemented. Thus, the associated time to the processing of the optimum number of signals for the recognition of gait events is called decision time. For example, if $n_s = 3$, a number of three new inputs is computed, where inputs $\hat{x}_1 = \sum_{k=1}^{3} x_k/n_s$, $\hat{x}_2 = \sum_{k=4}^{7} x_k/n_s$ and $\hat{x}_4 = x_{25}$. In Fig. 11, the average testing accuracy of ten random experiments obtained by the ML-IT2-FELM and KML-IT2-FELM using six different values for n_s is illustrated.

As can be noted from Fig. 11, the highest average testing performance for the ML-IT2-FELM is achieved by using a value for $n_s = 25$, while for the KML-IT2-FELM is $n_s = 1$. According to our simulations, the highest trade-off between testing accuracy and decision time is achieved by using $n_s = 5$ (105 ms) for the ML-IT2-FELM and $n_s = 1$ (94 ms) for the KML-IT2-FELM. From Fig. 11, it is also clear that for both models, the lowest accuracy is obtained when $n_s = 3$.

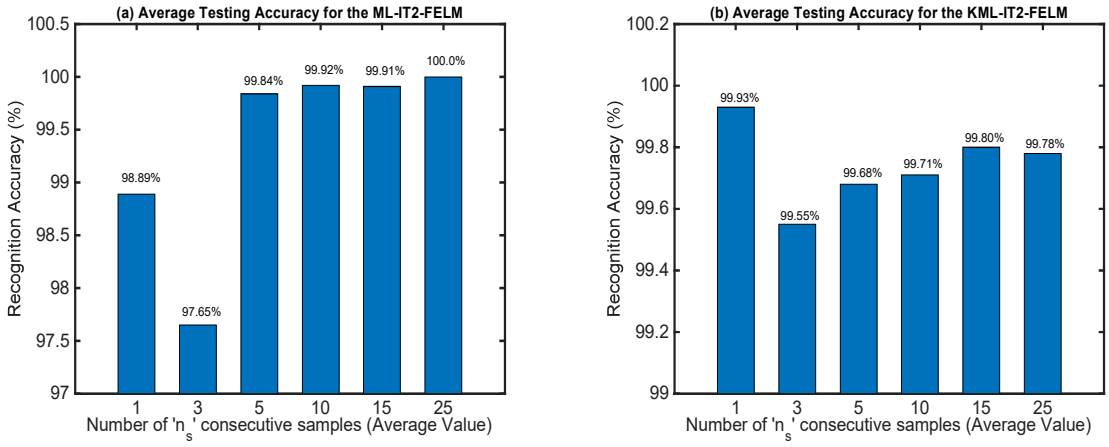

Figure 11: Average Testing accuracy for the recognition of Gait Events of six experiments using a different average value for the (a) ML-IT2-FELM and (b) KML-IT2-FELM.

To illustrate the statistical variability of the suggested models, in Fig. 12, a box-and-whisker diagram that corresponds to the testing accuracy of the ML-IT2-FELM and KML-IT2-FELM with an average number n_s of 5 and 1 contiguous signals is presented respectively.

According to Fig. 12(a), a clear model variability is produced when using a value of $n_s < 5$ for the ML-IT2-FELM. Opposite to this, from Fig. 12(b), model accuracy for the recognition of gait events using a KML-IT2-FELM is significantly affected when $n_s > 15$ contiguous signal values. To identify model contributions of the ML-IT2-FELM and KML-IT2-FELM in relation to other ML techniques such as ML-ELM, CNN, MML-KELM and an ML-IT2-FELM with a Nie-Tan type-reduction process at each hidden layer, in Fig. 13 a box-and-whisker plot is presented. To evaluate the model distribution provided in Fig. 13, a one-way ANOVA procedure with a significance value $\alpha = 0.001$ was implemented. From Fig.13, even a certain level of overlapping among the means of all ML techniques, a clear variance in model accuracy influenced by choice of ML learning technique is contained. Particularly those solutions produced by CNN, ML-IT2-FELM-NT and ML-KELM in relation to ML-ELM, ML-IT2-FELM and KML-IT2-FELM.

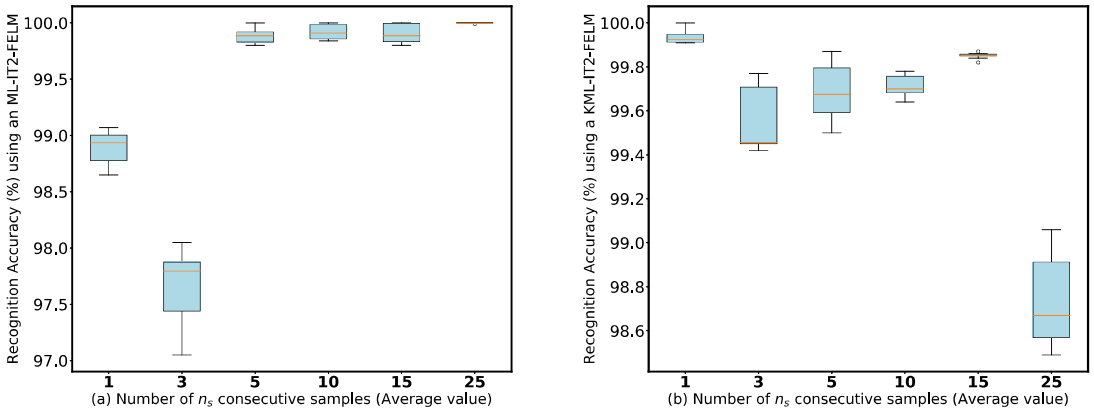

Figure 12: Box-and-whisker plots for the recognition of gait events using an (a) ML-IT2-FELM and (b) KML-IT2-FELM with a value for $n_s = 1, 3, 5, 10, 15, 25$.

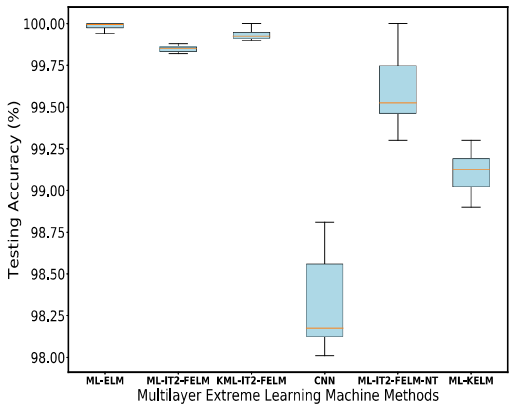

Figure 13: Box-and-whisker plots for the recognition of gait events using six different ML neural structures.

Table 2: One-way ANOVA results for the recognition of gait events.

Source	SS	df	MS	F-value	P-value
Model Accuracy	20.823	5	4.165	161.7	3.3229e-31
Within	1.390	54	0.025		
Total	22.21	59			

Moreover, it is also evident no outlier solutions were produced by any ML methodology either to recognise walking activities or gait events. For the recognition of individual gait events, the confusion matrices in Fig. 14 show the average training and testing accuracy provided by the ML-IT2-FELM and KML-IT2-FELM using 5 contiguous samples respectively.

In this experiment, the network structure of the ML-IT2-FELM involves 3 hidden layers, $[500 - 200 - 200 - 300 - 8]$. Opposite to ML-IT2-FELM, finding the appropriate number of M fuzzy rules for the KML-IT2-FELM is not necessary. The regularisation parameters C_i used by the ML-IT2-FELM and KML-IT2-FELM is defined as $C_i = [0.15, 9.1 \times 10^2, 4 \times 10^2, 820]$ and $C_i = [1.1, 3.4 \times 10^2, 1 \times 10^4, 190]$. To perform each experiment, the original IMU data set was normalised to the interval $[0 - 1]$. The confusion matrix of the average training and testing of ten random experiments that correspond to the ML-IT2-FELM and KML-IT2-FELM is presented in Fig. 14, where black and white colours represent low and high accuracy respectively.

Similar to the recognition of walking activities, the first four gait events are slightly affected by noisy measurements not only during the training stage but also to recognise unseen signals. As illustrated in Fig. 14, the proposed ML strategies are more accurate for the prediction of gait events that correspond to the swing phase, in particular to mid and terminal swings. These experimental results show not only the ability and robustness to recognise gait events in the presence of noise, but also to determine in which gait phase is the human during the walking activity, i.e. stance or swing phase.

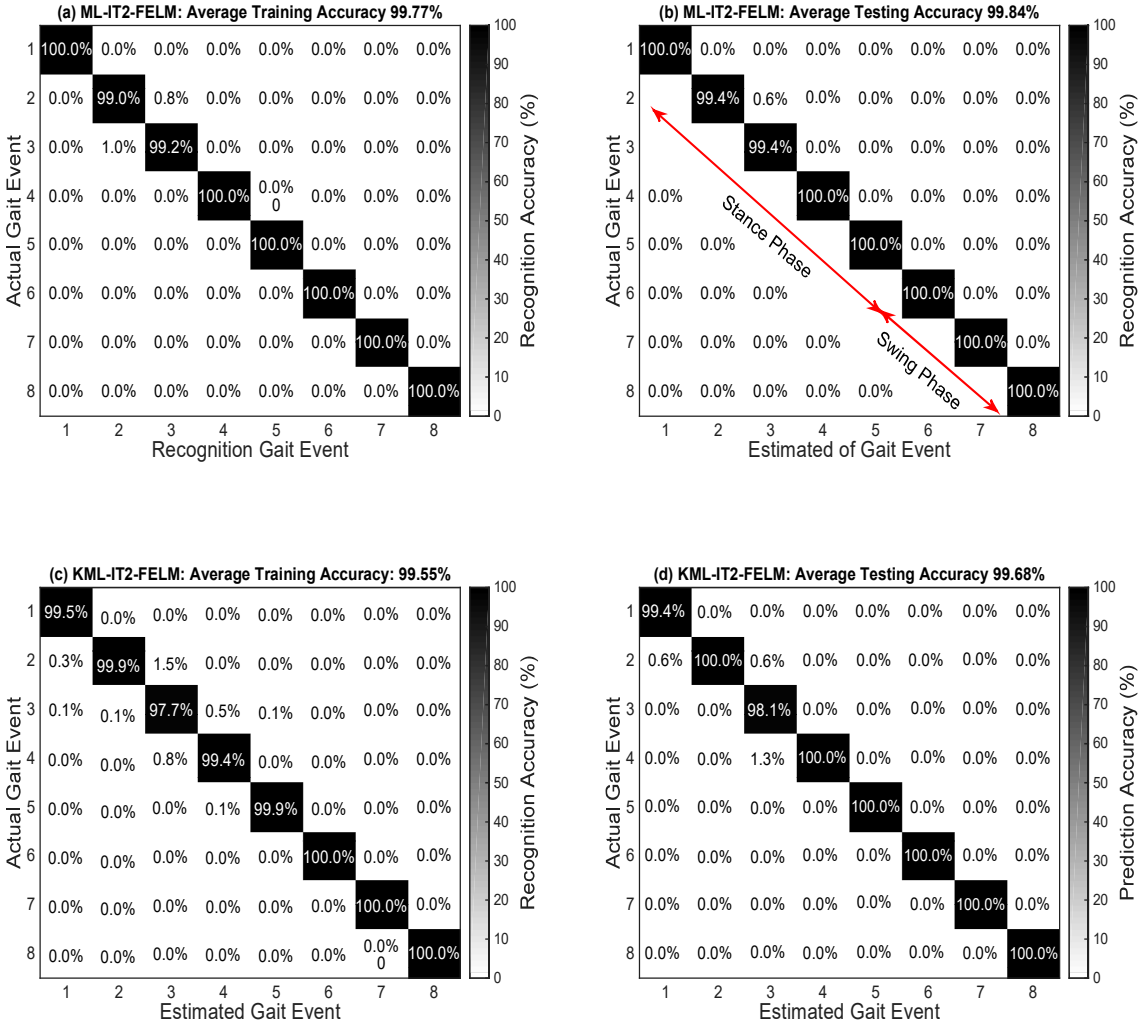

Figure 14: Training and testing confusion matrix obtained by the ML-IT2-FELM and KML-IT2-FELM to the recognition of stance and swing phases composed of eight gait events: (1) initial contact, (2) loading response, (3) mid stance, (4) terminal stance, (5) pre-swing, (6) initial swing, (7) mid swing and (8) terminal swing.

5.3. Performance analysis with respect to state-of-art shallow and multilayer strategies

This section provides a comparison analysis of the average testing performance obtained by the proposed ML-IT2-FELM, KML-IT2-FELM, a ML-IT2-FELM using an IT2-FELM with a NT direct defuzzification as the final layer (ML-IT2-FELM-NT for short) and other existing shallow and multilayer strategies to the recognition of walking activities and gait events. As detailed in previous sections, the robustness of all the methodologies presented in Table 1 are tested adding Gaussian noise with signal-to-noise ratio of 50 dB. Particularly, the noise was randomly added to all sensors for each decision process along the walking cycle. As can be observed from Table 1, the recognition of gait events is affected by the added noise. In table 1, column 'accuracy' indicates the average testing performance that results from ten random experiments while column 'decision time' is the average number of milliseconds that is required to recognise either a walking activity or a gait event. In other words, decision time is the optimum number of contiguous samples (signal measurements) necessary to recognise a human walking activity. Based on our simulation

results, the optimum number of contiguous samples for the ML-IT2-FELM and KML-IT2-FELM that provides the highest trade-off between model accuracy and decision time for the recognition of walking activities and gait events is 5 and 1 respectively.

According to Table 1, for the recognition of walking activities, the proposed ML structures as well as the ML-ELM and the ML-IT2-FELM-NT provided the highest trade-off between accuracy and low computational load on the one hand. On the other hand, for the recognition of gait events, ML-ELM and the proposed ML-IT2-FELM and KML-IT2-FELM outperformed not only other ML strategies, but also other shallow structures such as ELM, ANN [53], SVM [51] and an ensemble of classifiers [49]. Thus, the implementation of ML ELM theory improves not only the performance of existing IT2 fuzzy methodologies, but also the original ELM. As the proposed ML methodologies do not use input weights to compute the firing strengths at each FAE, an input projection mechanism is not necessary. This accounts to reduce the final decision time while providing a high model performance in terms of generalisation. In this sense, KML-IT2-FELM shares the ability of Kernel-based ELM to avoid manual tuning of the number of hidden units (fuzzy rules), all transformation matrices are unified into two matrices where each transformation is also learned by exact inverse, and as presented in Table 1, decision time is particularly reduced.

Interestingly, using an IT2-FELM as a classification layer, the robustness of the proposed ML strategies is higher than that provided by other methodologies such as the ML-IT2-FELM-NT, ML-KELM, ELM and RBFNN. This confirms the ability of IT2 Fuzzy Logic Systems to better deal with uncertainty, in particular to the classification of walking activities in the presence of noisy signals.

6. Conclusions

In this research work, a novel Multilayer Interval Type-2 Fuzzy Extreme Learning Machine (ML-IT2-FELM) for the recognition of walking activities and gait events was presented. The ML-IT2-FELM is a neural structure that consists of two main components. First, a number of stacked Interval Type-2 Fuzzy Autoencoders (FAEs) for unsupervised learning and feature extraction was implemented. The second component is an IT2-FELM that was employed for the classification of the extracted features. To compare the effectiveness of the proposed ML-IT2-FELM with respect to other ML fuzzy approaches, an ML-IT2-FELM method, inspired by kernel learning and called KML-IT2-FELM was also implemented. To evaluate the model performance of the ML-IT2-FELM and KML-IT2-FELM, two different experiments were used. In the first experiment, both models were applied to the recognition of three different walking activities, i.e. 1) Level-Ground Walking (LGW), 2) Ramp Ascent (RA) and 3) Ramp Descent (RD), while the second experiment is about the recognition of eight different gait events.

The results for the recognition of walking activities and gait events showed that an average accuracy of 99.97% and 99.84%, with decision times of 290.4ms and 105ms, were achieved by the ML-IT2-FELM, while the KML-IT2-FELM provided an average accuracy of 99.98% and 99.93% with a decision times of 191.9ms and 94ms. These results allow us to conclude the following:

Table 3: Activity and gait recognition performance between the ML-IT2-FELM, KML-IT2-FELM and with respect to other shallow and multilayer methodologies.

Model	**Activity**	**# Sensors**	**Recognition Activity**		**Recognition Gait Event**	
			Accuracy (%)	Decision time(ms)	Accuracy (%)	Decision time(ms)
ELM	LGW, RA, RD	3	99.66	54.1	99.19	32.3
SVM [51]	LGW, RA, RD, stair	9	99.00	150.4	97.00	-
DBN [50]	LGW, RA, RD, stairs	13	98.00	300.9	95.25	-
ANN [53]	LGW	3	98.78	-	-	-
Adaptive BasIS [11]	LGW, RA, RD	3	99.87	240.0	99.20	130.0
CNN [5]	LGW, RA, RD	3	**99.88**	380.3	98.32	279.3
Ensemble of Classifiers [49]	LGW, RA, RD, stairs	9	97.60	-	-	-
ELM-RBFNN [30]	LGW, RA, RD	3	99.83	70.1	99.11	34.2
RBFNN [30]	LGW, RA, RD	3	98.83	210.3	97.00	40.6
IT2-RBFNN [40]	LGW, RA, RD	3	99.59	240.1	98.71	72.8
ML-ELM [28]	LGW, RA, RD	3	**99.97**	**124.3**	**99.99**	**83.9**
ML-IT2-FELM	LGW, RA, RD	3	**99.97**	**290.4**	**99.84**	**105.0**
ML-IT2-FELM-NT	LGW, RA, RD	3	**99.96**	**156.5**	99.59	67.2
IT2-ELM-NT	LGW, RA, RD	3	99.77	70.2	98.92	38.2
IT2-ELM-KM [30]	LGW, RA, RD	3	99.88	52.9	99.30	45.0
KML-IT2-FELM	LGW, RA, RD	3	**99.98**	**191.9**	**99.93**	**94.0**
ML-KELM [18]	LGW, RA, RD	3	99.88	81.7	99.10	65.0

- The ML-IT2-FELM and the KML-IT2-FELM do not need an input projection mechanism as each FAE is based on the IT2-RBFNN.
- Both models achieve a more robust and better model accuracy compared to other state-of-art approaches.
- The KML-IT2-FELM avoids manual tuning of the number of hidden nodes or fuzzy rules and each transformation is also learned by exact inverse.
- Both ML models follow a hierarchical learning scheme where parameters of each hidden layer do not need to be tuned.

7. References

[1] S. C. Mukhopadhyay, Wearable sensors for human activity monitoring: A review, IEEE sensors journal 15 (3) (2015) 1321–1330.

[2] O. D. Lara, A. J. Pérez, M. A. Labrador, J. D. Posada, Centinela: A human activity recognition system based on acceleration and vital sign data, Pervasive and mobile computing 8 (5) (2012) 717–729.

[3] U. Martinez-Hernandez, A. A. Dehghani-Sanij, Probabilistic identification of sit-to-stand and stand-to-sit with a wearable sensor, Pattern Recognition Lettersdoi:https://doi.org/10.1016/j.patrec.2018.03.020.

[4] M. Zhang, A. A. Sawchuk, Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, 2012, pp. 1036–1043.

[5] U. Martinez-Hernandez, A. Rubio-Solis, A. A. Dehghani-Sanij, Recognition of walking activity and prediction of gait periods with a cnn and first-order mc strategy, in: 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob), IEEE, 2018, pp. 897–902.

[6] F. Sikder, D. Sarkar, Log-sum distance measures and its application to human-activity monitoring and recognition using data from motion sensors, IEEE Sens. J 17 (2017) 4520–4533.

[7] M. Chen, Y. Li, X. Luo, W. Wang, L. Wang, W. Zhao, A novel human activity recognition scheme for smart health using multilayer extreme learning machine, IEEE Internet of Things Journal.

[8] O. D. Lara, M. A. Labrador, et al., A survey on human activity recognition using wearable sensors., IEEE Communications Surveys and Tutorials 15 (3) (2013) 1192–1209.

[9] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, Y. Amirat, Physical human activity recognition using wearable sensors, Sensors 15 (12) (2015) 31314–31338.

[10] M. Zhang, A. A. Sawchuk, Human daily activity recognition with sparse representation using wearable sensors, IEEE journal of Biomedical and Health Informatics 17 (3) (2013) 553–560.

[11] U. Martinez-Hernandez, A. A. Dehghani-Sanij, Adaptive bayesian inference system for recognition of walking activities and prediction of gait events using wearable sensors, Neural Networks 102 (2018) 107–119.

[12] U. Martinez-Hernandez, T. J. Dodd, M. H. Evans, T. J. Prescott, N. F. Lepora, Active sensorimotor control for tactile exploration, Robotics and Autonomous Systems 87 (2017) 15–27.

[13] W.-Y. Deng, Q.-H. Zheng, Z.-M. Wang, Cross-person activity recognition using reduced kernel extreme learning machine, Neural Networks 53 (2014) 1–7.

[14] D. Wu, Z. Wang, Y. Chen, H. Zhao, Mixed-kernel based weighted extreme learning machine for inertial sensor based human activity recognition with imbalanced dataset, Neurocomputing 190 (2016) 35–49.

[15] A. Rubio-Solis, G. Panoutsos, An ensemble data-driven fuzzy network for laser welding quality prediction, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–6.

[16] M. Nie, W. W. Tan, Towards an efficient type-reduction method for interval type-2 fuzzy logic systems, in: Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on, IEEE, 2008, pp. 1425–1432.

[17] D. Wu, J. M. Mendel, Enhanced karnik–mendel algorithms, IEEE Transactions on Fuzzy Systems 17 (4) (2009) 923–934.

[18] C. M. Wong, C. M. Vong, P. K. Wong, J. Cao, Kernel-based multilayer extreme learning machines for representation learning, IEEE transactions on neural networks and learning systems 29 (3) (2018) 757–762.

[19] Z. Wang, D. Wu, R. Gravina, G. Fortino, Y. Jiang, K. Tang, Kernel fusion based extreme learning machine for cross-location activity recognition, Information Fusion 37 (2017) 1–9.

[20] L. Hu, Y. Chen, J. Wang, C. Hu, X. Jiang, Okrelm: online kernelized and regularized extreme learning machine for wearable-based activity recognition, International Journal of Machine Learning and Cybernetics 9 (9) (2018) 1577–1590.

[21] A. Malaisé, P. Maurice, F. Colas, F. Charpillet, S. Ivaldi, Activity recognition with multiple wearable sensors for industrial applications, in: Advances in Computer-Human Interactions, 2018.

[22] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, U. R. Alo, Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges, Expert Systems with Applications.

[23] N. Y. Hammerla, S. Halloran, T. Ploetz, Deep, convolutional, and recurrent models for human activity recognition using wearables, arXiv preprint arXiv:1604.08880.

[24] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, J. Zhang, Convolutional neural networks for human activity recognition using mobile sensors, in: Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on, IEEE, 2014, pp. 197–205.

[25] L. Wang, Recognition of human activities using continuous autoencoders with wearable sensors, Sensors 16 (2) (2016) 189.

[26] C. A. Ronao, S.-B. Cho, Human activity recognition with smartphone sensors using deep learning neural networks, Expert Systems with Applications 59 (2016) 235–244.

[27] L. L. C. Kasun, H. Zhou, G.-B. Huang, C. M. Vong, Representational learning with extreme learning machine for big data, IEEE intelligent systems 28 (6) (2013) 31–34.

[28] J. Tang, C. Deng, G.-B. Huang, Extreme learning machine for multilayer perceptron, IEEE transactions on neural networks and learning systems 27 (4) (2016) 809–821.

[29] Y. Yang, Q. J. Wu, Multilayer extreme learning machine with subnetwork nodes for representation learning, IEEE transactions on cybernetics 46 (11) (2016) 2570–2583.

[30] A. Rubio-Solis, U. Martinez-Hernandez, G. Panoutsos, Evolutionary extreme learning machine for the interval type-2 radial basis function neural network: A fuzzy modelling approach, in: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2018, pp. 1–8.

[31] C.-M. Vong, C. Chen, P.-K. Wong, Empirical kernel map-based multilayer extreme learning machines for representation learning, Neurocomputing.

[32] P. Vidnerová, R. Neruda, Deep networks with rbf layers to prevent adversarial examples, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2018, pp. 257–266.

[33] M. D. Tissera, M. D. McDonnell, Deep extreme learning machines: supervised autoencoding architecture for classification, Neurocomputing 174 (2016) 42–49.

[34] L. L. C. Kasun, Y. Yang, G.-B. Huang, Z. Zhang, Dimension reduction with extreme learning machine, IEEE Transactions on Image Processing 25 (8) (2016) 3906–3918.

[35] U. Martinez-Hernandez, A. Rubio-Solis, G. Panoutsos, A. A. Dehghani-Sanij, A combined adaptive neuro-fuzzy and bayesian strategy for recognition and prediction of gait events using wearable sensors, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–6.

[36] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, L. Ljung, Kernel methods in system identification, machine learning and function estimation: A survey, Automatica 50 (3) (2014) 657–682.

[37] M. Pal, A. E. Maxwell, T. A. Warner, Kernel-based extreme learning machine for remote-sensing image classification, Remote Sensing Letters 4 (9) (2013) 853–862.

[38] C.-F. Juang, R.-B. Huang, W.-Y. Cheng, An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems, IEEE Transactions on fuzzy systems 18 (4) (2010) 686–699.

[39] R. Chimatapu, H. Hagras, A. Starkey, G. Owusu, Interval type-2 fuzzy logic based stacked autoencoder deep neural network for generating explainable ai models in workforce optimization, in: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2018, pp. 1–8.

[40] A. Rubio-Solis, G. Panoutsos, Interval type-2 radial basis function neural network: A modeling framework, IEEE Transactions on Fuzzy Systems 23 (2) (2015) 457–473.

[41] A. R. Solis, G. Panoutsos, Granular computing neural-fuzzy modelling: A neutrosophic approach, Applied Soft Computing 13 (9) (2013) 4010–4021.

[42] A. Rubio-Solis, P. Melin, U. Martinez-Hernandez, G. Panoutsos, General type-2 radial basis function neural network: A data-driven fuzzy model, IEEE Transactions on Fuzzy Systems 27 (2) (2018) 333–347.

[43] A. Rubio-Solis, G. Panoutsos, S. Thornton, A data-driven fuzzy modelling framework for the classification of imbalanced data, in: Intelligent Systems (IS), 2016 IEEE 8th International Conference on, IEEE, 2016, pp. 302–307.

[44] J. M. Mendel, General type-2 fuzzy logic systems made simple: a tutorial, IEEE Transactions on Fuzzy Systems 22 (5) (2014) 1162–1182.

[45] J. M. Mendel, Computing derivatives in interval type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 12 (1) (2004) 84–98.

[46] A. Rubio-Solis, G. Panoutsos, Fuzzy uncertainty assessment in rbf neural networks using neutrosophic sets for multiclass classification, in: 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2014, pp. 1591–1598.

[47] D. Wu, Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons, IEEE Transactions on Fuzzy Systems 21 (1) (2013) 80–99.

[48] A. Rubio-Solis, A. Baraka, G. Panoutsos, S. Thornton, Data-driven interval type-2 fuzzy modelling for the classification of imbalanced data, in: Practical Issues of Intelligent Innovations, Springer, 2018, pp. 37–51.

[49] E. Fullerton, B. Heller, M. Munoz-Organero, Recognising human activity in free-living using multiple body-worn accelerometers, IEEE Sensors Journal 17 (16) (2017) 5290–5297.

[50] A. J. Young, A. M. Simon, N. P. Fey, L. J. Hargrove, Intent recognition in a powered lower limb prosthesis using time history information, Annals of biomedical engineering 42 (3) (2014) 631–641.

[51] H. Huang, F. Zhang, L. J. Hargrove, Z. Dou, D. R. Rogers, K. B. Englehart, Continuous locomotion-mode identification for prosthetic legs based on neuromuscular–mechanical fusion, IEEE Transactions on Biomedical Engineering 58 (10) (2011) 2867–2875.

[52] U. Martinez-Hernandez, A. A. Dehghani-Sanij, Probabilistic identification of sit-to-stand and stand-to-sit with a wearable sensor, Pattern Recognition Letters 118 (2019) 32–41.

[53] K. Zhang, M. Sun, D. K. Lester, F. X. Pi-Sunyer, C. N. Boozer, R. W. Longman, Assessment of human locomotion by using an insole measurement system and artificial neural networks, Journal of biomechanics 38 (11) (2005) 2276–2287.